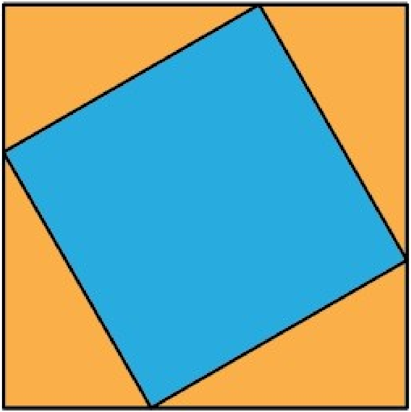
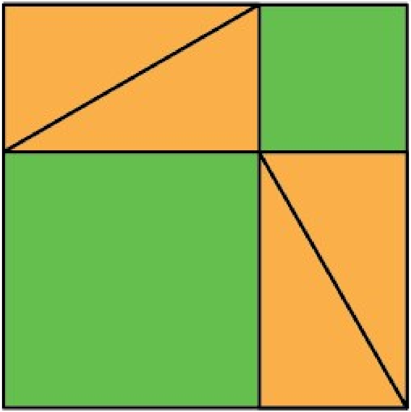
What are...computer proofs?

Or: Who needs mathematicians?

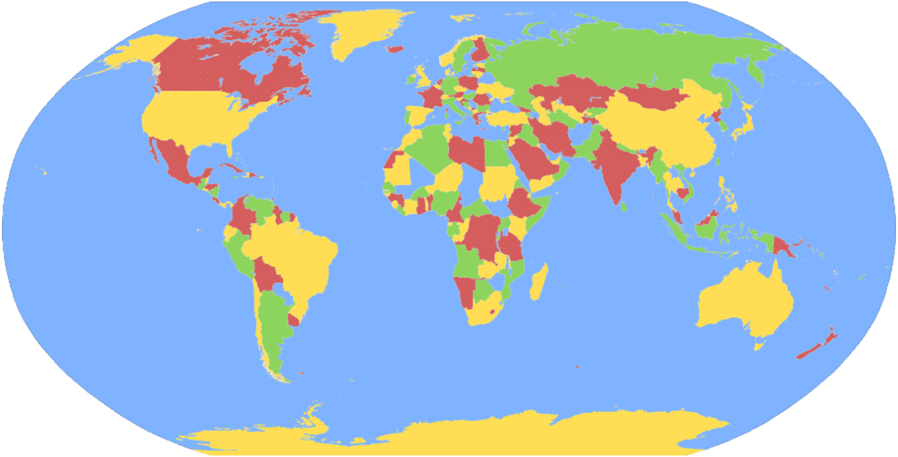
A proof





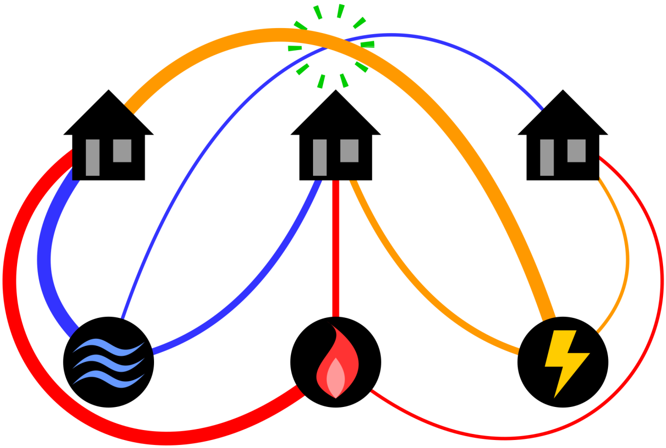
- The above is a proof
- What is a proof?
- I will not really answer that ;-) so: “proof = whatever is accepted as a proof”

Stage 1: computer assisted proofs



- Computers are very often used to verify "ugly" parts of proofs
- Example A difficult integral solved by a computer algebra system
- Example A large case-by-case check that is too large to do by hand

Stage 2: computer verified proofs



- Computers are sometimes used to verify proofs
- Example Quadratic reciprocity has a computer verified proof
- Example The Jordan curve theorem has a computer verified proof

Enter, the theorem

The following (and more) proofs have been computer verified:

Year	Theorem	Proof System	Formalizer	Traditional Proof
1986	First Incompleteness	Boyer-Moore	Shankar	Gödel
1990	Quadratic Reciprocity	Boyer-Moore	Russinoff	Eisenstein
1996	Fundamental - of Calculus	HOL Light	Harrison	Henstock
2000	Fundamental - of Algebra	Mizar	Milewski	Brynski
2000	Fundamental - of Algebra	Coq	Geuvers et al.	Kneser
2004	Four-Color	Coq	Gonthier	Robertson et al.
2004	Prime Number	Isabelle	Avigad et al.	Selberg-Erdös
2005	Jordan Curve	HOL Light	Hales	Thomassen
2005	Brouwer Fixed Point	HOL Light	Harrison	Kuhn
2006	Flyspeck I	Isabelle	Bauer-Nipkow	Hales
2007	Cauchy Residue	HOL Light	Harrison	classical
2008	Prime Number	HOL Light	Harrison	analytic proof

The Formal Jordan Curve Theorem

$$\begin{aligned}&\forall C.\ simple\_closed\_curve\ top2\ C \Rightarrow\\ &\quad (\exists A\,B.\ top2\ A \wedge top2\ B \wedge\\ &\qquad connected\ top2\ A \wedge connected\ top2\ B \wedge\\ &\qquad A \neq \emptyset \wedge B \neq \emptyset \wedge\\ &\qquad A \cap B = \emptyset \wedge A \cap C = \emptyset \wedge B \cap C = \emptyset \wedge\\ &\qquad A \cup B \cup C = euclid\ 2\ )\end{aligned}$$

- I am slow... This list is from 2008 – much more has been done!
- A key step is to put statements into computer readable form

Stage 3: automated proofs

Full Automation of the Robbins Conjecture

Let S be a nonempty set with an associative commutative binary operation $(x, y) \mapsto xy$ and a unary operation $x \mapsto [x]$ (which, for convenience, we write synonymously as $x \mapsto \bar{x}$). The Robbins conjecture (in Winker form) asserts that the general Robbins identity

$$[[ab][a\bar{b}]] = a$$

implies the existence of $c, d \in S$ such that $[cd] = \bar{c}$. Here is the original proof that EQN discovered, as reconstructed in [10].

Proof. A solution is $c = x^3u, d = xu$, where $u = [x\bar{x}]$ and x is arbitrary. Abbreviate $j = [cd]$, $e = u[x^2]\bar{c}$. Over the equality sign, a prime indicates a direct application of the Robbins identity; a superscript indicates a substitution of the numbered line; no superscript indicates a rewriting of abbreviations c, d, e, j, u.

$$\begin{aligned}
0 : [u[x^2]] \quad & = [[x\bar{x}][xx]] =' x. \\
1 : [xu[xu[x^2]\bar{c}]] \quad & =' [[[xux^2][xu[x^2]]][xu[x^2]\bar{c}]] = [[\bar{c}[xu[x^2]]][\bar{c}xu[x^2]]] =' \bar{c}. \\
2 : [u\bar{c}] \quad & = [u[x^2ux]] =^{0} [u[x^2u[u[x^2]]]] =' [[[ux^2][u[x^2]]][x^2u[u[x^2]]]] \\
& =' [u[x^2]] =^{0} x. \\
3 : [ju] \quad & = [[xcu]u] =' [[xcu][[uc][u\bar{c}]]] =^{2} [[xcu][x[cu]]] =' x \\
4 : [x[x[x^2]u\bar{c}]] \quad & =' [[[x[u\bar{c}]][xu\bar{c}]][x[x^2]u\bar{c}]] =^{2} [[[x^2][xu\bar{c}]][[x^2]xu\bar{c}]] =' [x^2] \\
5 : [x\bar{c}] \quad & =^{1} [x[xu[xu[x^2]\bar{c}]]] =^{0} [[u[x^2]][xu[xu[x^2]\bar{c}]]] \\
& = [[u[x^2]][ux[xe]]] =^{4} [[u[x[xe]]][ux[xe]]] =' u \\
6 : [jx] \quad & =' [j[[xc][x\bar{c}]]] =^{5} [j[[xc]u]] = [[uxc][u[xc]]] =' u \\
7 : [cd] \quad & = j =' [[j[x\bar{c}]][jx\bar{c}]] =^{5} [[ju][jx\bar{c}]] =^{3} [x[jx\bar{c}]] =^{2} [[\bar{c}u][\bar{c}jx]] \\
& =^{6} [[\bar{c}[jx]][\bar{c}jx]] =' \bar{c}.
\end{aligned}$$

□

- Computers should be used more often to proof new theorems
- Example Robbins conjecture (a certain conjecture in universal algebra) – many people tried to prove it but only a computer managed to do it!

Thank you for your attention!

I hope that was of some help.